**Shahzaman Abbasi**

**023-20-012**

**Lab – 06 Exercises**

**Task 1:**

**Code:**

class Medicine {

String medicineName, strength;

int quantity;

Medicine(String medicineName, String strength, int quantity) {

this.medicineName = medicineName;

this.strength = strength;

this.quantity = quantity;

}

public void displayDetails() {

System.out.println("Medicine Name: " + medicineName);

System.out.println("Strength: " + strength);

System.out.println("Quantity: " + quantity);

}

public int getQuantity() {

return quantity;

}

public void setQuantity(int quantity) {

this.quantity = quantity;

}

}

public class MedicalInventoryManager {

public static void updateMedicineQuantity(Medicine medicine, int quantityToAdd) {

int currentQuantity = medicine.getQuantity();

medicine.setQuantity(currentQuantity + quantityToAdd);

}

public static void displayDetails(Medicine medicine) {

medicine.displayDetails();

}

public static void main(String[] args) {

Medicine medicine1 = new Medicine("Amoxicillin", "250mg", 80);

Medicine medicine2 = new Medicine("Ibuprofen", "200mg", 60);

System.out.println("Medicine Details before update:");

displayDetails(medicine1);

displayDetails(medicine2);

updateMedicineQuantity(medicine1, 50);

updateMedicineQuantity(medicine2, 20);

System.out.println("\nMedicine Details after update:");

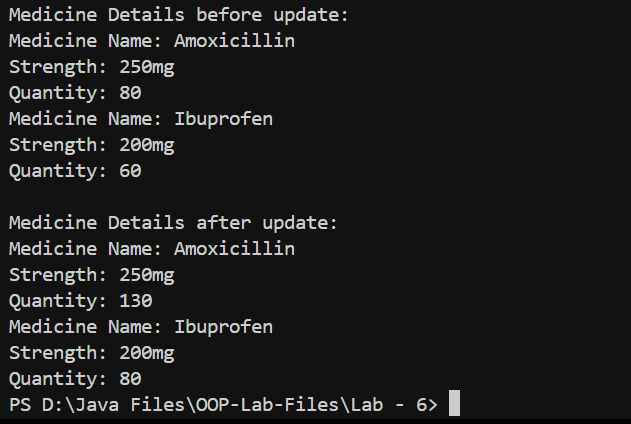
displayDetails(medicine1);

displayDetails(medicine2);

}

}

**Output:**

****

**Task 2:**

**Code:**

import java.util.Scanner;

class Student {

int id;

String name;

int age;

public Student(int id, String name, int age) {

this.id = id;

this.name = name;

this.age = age;

}

public int getId() {

return id;

}

public String getName() {

return name;

}

public int getAge() {

return age;

}

}

public class StudentDatabase {

static Student[] students = {

new Student(1, "Shahzaman", 21),

new Student(2, "Deepika", 21),

new Student(3, "Yawar", 21),

new Student(4, "Simran", 20)

};

static Student searchStudentById(int ID) {

for (Student student : students) {

if (student.getId() == ID) {

return student;

}

}

return null;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the ID of the student to search:");

int searchId = scanner.nextInt();

Student foundStudent = searchStudentById(searchId);

if (foundStudent != null) {

System.out.println("Student found!");

System.out.println("ID: " + foundStudent.getId());

System.out.println("Name: " + foundStudent.getName());

System.out.println("Age: " + foundStudent.getAge());

} else {

System.out.println("Student not found.");

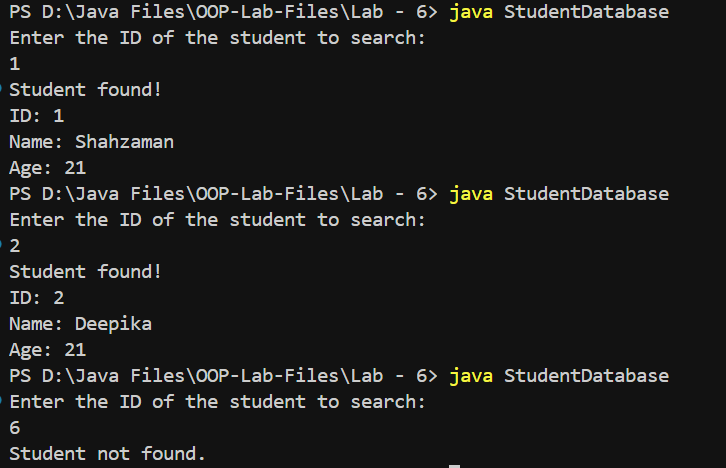
}

scanner.close();

}

}

**Output:**

****

**Task 3:**

Write a method that returns the sum of all even Fibonacci numbers using recursion. Consider

all Fibonacci numbers that are less than or equal to n.

**Code:**

public class EvenFibSum {

static int evenFibonacciSum(int n) {

if (n <= 0) {

return 0;

} else if (n == 1) {

return 0;

} else if (n == 2) {

return 2;

}

return evenFibonacciSum(n, 1, 2, 2);

}

static int evenFibonacciSum(int n, int num1, int num2, int sumOfEven) {

int fib = num1 + num2;

if (fib > n) {

return sumOfEven;

}

if (fib % 2 == 0) {

sumOfEven += fib;

}

return evenFibonacciSum(n, num2, fib, sumOfEven);

}

public static void main(String[] args) {

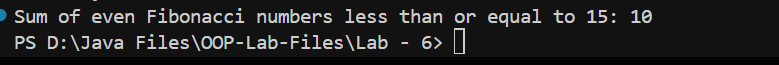
int n = 15;

System.out.println("Sum of even Fibonacci numbers less than or equal to " + n + ": " + evenFibonacciSum(n));

}

}

**Output:**



**Task 4:**

Write a recursive function to compute power of a number (X^n). Test and trace for 45

**Code:**

public class PowerOfNum {

public static double power(double x, int n) {

if (n == 0) {

return 1;

} else if (n > 0) {

return x \* power(x, n - 1);

} else {

return (1 / x) \* power(x, n + 1);

}

}

public static void main(String[] args) {

double x = 4;

int n = 5;

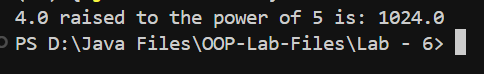
double result = power(x, n);

System.out.println(x + " raised to the power of " + n + " is: " + result);

}

}

**Output:**

****